Greedy Maximal Scheduling in Wireless Networks

Qiao Li
qiaoli@cmu.edu
Department of Electrical and Computer Engineering
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213

Rohit Negi
negi@ece.cmu.edu
Department of Electrical and Computer Engineering
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, PA 15213

***Abstract*—In this paper we consider greedy scheduling algorithms in wireless networks, i.e., the schedules are computed by adding links greedily based on some priority vector. Two special cases are considered: 1) Longest Queue First (LQF) scheduling, where the priorities are computed using queue lengths, and 2) Static Priority (SP) scheduling, where the priorities are pre-assigned. We first propose a closed-form lower bound stability region for LQF scheduling, and discuss the tightness result in some scenarios. We then propose an lower bound stability region for SP scheduling with multiple priority vectors, as well as a heuristic priority assignment algorithm, which is related to the well-known Expectation-Maximization (EM) algorithm. The performance gain of the proposed heuristic algorithm is finally confirmed by simulations.**

I. Introduction

Optimal scheduling in wireless networks is, in general, an NP-complete problem. Essentially, in order to achieve the optimal stability region, one either needs to solve an NP-complete problem in each time slot (see the max-weight scheduling in [1]), or approach the optimal solution gradually over time slots (see the linear random scheduler in [2], and the CSMA-type scheduler in [3]), thereby reducing the computation complexity in each time slot by amortization over a relatively long period of time. The first case is clearly not practical, due to the high computation complexity. For the second case, the delay performance may be quite bad because the queue lengths can become very large even with low traffic, since, intuitively, it takes an exponential number of time slots to converge to a (near) optimal solution.

Recently, there have been significant research activities on sub-optimal scheduling algorithms with provable performance guarantees. In [4], maximal scheduling was proposed as a low (linear) complexity algorithm for wireless networks. In maximal scheduling, the only constraint is that the scheduled set of links is *maximal*, i.e., no more link can be added to the schedule without violating the interference constraint. It has been shown [4] that maximal scheduling can achieve a constant approximation ratio in typical wireless networks, which is the fraction of the optimal stability region that can be stabilized by maximal schedulers. Further, the delay performance of maximal scheduling is quite good under light traffic. However, since the class of maximal schedulers is broad, the worst case performance guarantee of maximal scheduling, in the form of a lower bound stability region, is pessimistic [4]. Thus, it is necessary to consider specific maximal schedulers for improved performance guarantees.

In this paper we consider two specific types of maximal schedulers: Longest Queue First (LQF) scheduling and Static Priority (SP) scheduling. In LQF scheduling, the schedule is computed by queue length based priorities, i.e., links are added according to their queue lengths, from the longest to the shortest, and a link with non-empty queue is added to the schedule whenever there is no conflict. In the literature, it has been shown that LQF scheduling is optimal if the network satisfies the “local-pooling condition” [9], which is a function of the network topology. Later, it was generalized to the notion of “local-pooling factor” [5], which was shown to be equal to the approximation ratio. However, to the best of the authors’ knowledge, it is hard to specify the stability region of LQF scheduling. That is, given a vector of packet arrival rates, it is difficult to predict whether that rate can be supported by LQF scheduling, since checking the “local-pooling condition” requires complexity exponential in the network size n. This is particularly inconvenient for cross-layer optimization, where one needs to allocate the link rates efficiently, subject to stability constraints under LQF scheduling. In this paper, we propose a closed-form lower bound stability region for LQF scheduling, which is further shown to be tight in some scenarios. Further, we propose a fast (linear complexity) checking algorithm, which can decide whether a given arrival rate vector is inside the lower bound stability region.

We next consider SP scheduling, where the only difference from LQF scheduling is that the links to be added to a schedule are considered following (pre-computed) static priorities, instead of queue lengths. Thus, the implementation is simpler than LQF scheduling, where the changes in queue lengths often generate a considerable amount of messages to be exchanged across the network. Further, SP scheduling has comparable performance to LQF scheduling. For example, in [6] and [7] we have shown that the stability region of SP scheduling with a single priority can achieve the same lower bound stability region of LQF scheduling, and with $n+1$ priority vectors we can achieve the optimal stability region. In this paper, we try to analyze the performance of SP scheduling with an arbitrary number K of priority vectors. We first formulate a lower bound stability region for SP scheduling with K priority vectors, Next, we propose a heuristic priority assignment algorithm, which assigns two priority vectors using an Expectation-Maximization (EM) type

algorithm. This algorithm generalizes easily to K priorities. Finally, we demonstrate the performance gain of the SP scheduling through simulations.

The organization of this paper is as follows: In Section II we describe the queueing network model, In Section III we consider the performance of LQF scheduling. Section IV analyzes SP scheduling with multiple priorities, Section V demonstrate the simulation results. Finally, Section VI concludes this paper.

II. System Model

In this section we introduce the system model, which is standard in the literature.

A. Network Topology and Priority Vector

We consider the scheduling problem at the MAC layer of a wireless network, where the network topology is modeled as a conflict graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here, $\mathcal{V}$ is the set of n links, and $\mathcal{E}$ is the set of pairwise conflicts, i.e., $(i, j) \in \mathcal{E}$ implies that links i and j are not allowed to transmit simultaneously, due to the strong interference that one link can cause to the other. Note that this model is extensively used in the literature, and is suitable to model various physical layer constraints. For example, in Bluetooth or FH-CDMA networks, the only constraint is that a node can not both transmit and receive simultaneously. Thus, two links $(i, j) \in \mathcal{E}$ if and only if they share a common transceiver in the network. As another example, the ubiquitous 802.11 Distributed Coordination Function (DCF) implies that two links within two hops can not transmit together, due to the exchange of RTS/CTS messages. Therefore, two links $(i, j) \in \mathcal{E}$ if and only if they are within two-hop distance of each other.

For each link $i \in \mathcal{V}$, we define its neighborhood as $\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. We next introduce priority vectors, which are used later in the scheduling algorithm. A priority vector $\boldsymbol{p}$ is a $n \times 1$ vector which corresponds to a permutation of the vector $(1, 2, \ldots, n)^T$. Link i is said to have lower priority than link j if $p_i > p_j$. Thus, 1 is the highest priority, and n is the lowest priority. Given a priority vector $\boldsymbol{p}$, we define a priority weighted graph incidence matrix P, such that $P_{ii} = 1$, and $P_{ij} = \mathbf{1}_{\{j \in \mathcal{N}_i\}} \mathbf{1}_{\{p_i > p_j\}}$, where $\mathbf{1}_{\{\cdot\}}$ is the indicator function, i.e., $\mathbf{1}_{\{\text{true}\}} = 1$ and $\mathbf{1}_{\{\text{false}\}} = 0$. Thus, $P_{ij} = 1$ if and only if link j is a higher priority neighbor of link i.

B. Queueing Network

We assume that time is slotted, and associate each link i in the network with an external source $A_i(t)$, which is the cumulative packet arrival during the first t time slots. $\boldsymbol{A}(t)$ is the vectors of $A_i(t)$. The only constraints on the arrival process are that 1) it is uniformly bounded in each time slot, i.e., there exists a positive constant $0 < A_{\max} < \infty$, such that for all $t > 0$,

$$A_i(t) - A_i(t-1) \le A_{\max}, \ \forall i \in \mathcal{V} \tag{1}$$

and that 2) the arrival processes are subject to Strong Law of Large Numbers (SLLN), i.e., with probability 1 (w.p.1), we have

$$\lim_{t \to \infty} \boldsymbol{A}(t)/t = \boldsymbol{a} \tag{2}$$

where $\boldsymbol{a}$ is the average arrival rate vector. Note that this assumption on $\boldsymbol{A}(t)$ is quite mild, since it allows the processes $\boldsymbol{A}(t)$ to be correlated across time slots as well as across different links. Thus, it is suited to model practical packet sources, which are often subject to non-ergodic and correlating upper-layer mechanisms, such as routing and congestion control.

The queueing equation of the network is as the following

$$\boldsymbol{Q}(t) = \boldsymbol{Q}(0) + \boldsymbol{A}(t) - \boldsymbol{D}(t) \tag{3}$$

where $\boldsymbol{Q}(t)$ is the queue length vector at time slot t, and $\boldsymbol{D}(t)$ is the cumulative departure vector during the first t time slots. The departure vector at time slot t, which is denoted as $\Delta\boldsymbol{D}(t) \doteq \boldsymbol{D}(t) - \boldsymbol{D}(t-1)$, must correspond to an independent set in the conflict graph $\mathcal{G}$, so that, no packet contention happen.

We assume that the following scheduling produced departure vector $\Delta\boldsymbol{D}(t)$: In each time slot, the scheduler (either centralized or distributed) considers the links according to the sequence specified by a priority vector $\boldsymbol{p}(t)$, where $p_i(t)$ is the priority of link i during time slot t. Thus, a link with higher priority is always considered before the links with lower priorities. A link i under consideration is scheduled if and only if 1) it has nonempty queue and that 2) when link i is considered, none of the links in its neighborhood $\mathcal{N}_i$ have been scheduled. Note that, any greedy scheduler can be modeled in this way, with a proper choice of priority vector $\boldsymbol{p}(t)$. Specifically, LQF scheduling corresponds to computing $\boldsymbol{p}(t)$ by sorting the queue length vector $\boldsymbol{Q}(t-1)$, and SP scheduling corresponds to choosing a static $\boldsymbol{p}(t)$ sequence which are pre-computed priority vectors.

C. Stability Region

The performance of a scheduler is evaluated by its stability region, which is defined as the set of arrival rate vectors $\boldsymbol{a}$ such that *any arrival process* with average rate $\boldsymbol{a}$ is stable under the scheduler. We define stability as *rate stability* [8], i.e., $\lim_{t\to\infty} \boldsymbol{D}(t)/t = \boldsymbol{a}$ w.p.1. For specific stability regions, consider the following region: $\mathcal{A}_{\text{maximal}} = \{\boldsymbol{a} \in \mathbb{R}_+^n : a_i + \sum_{j \in \mathcal{N}_i} a_j \le 1, \forall i \in \mathcal{V}\}$. That is, the sum arrival rate in any link's neighborhood is no larger than 1. In [4], it has been shown that $\mathcal{A}_{\text{maximal}}$ can be stabilized by maximal scheduling. As another example, an SP scheduler with priority $\boldsymbol{p}$ can achieve the stability region [6] $\mathcal{A}_{\boldsymbol{p}} = \{\boldsymbol{a} \in \mathbb{R}_+^n : \|P\boldsymbol{a}\|_\infty \le 1\}$, where $\|\cdot\|_\infty$ is the infinity norm. We next illustrate these concepts with an example.

Example: Consider the 6-node conflict graph in Fig. 1(a), and assume that the priority is $\boldsymbol{p} = (1, 2, 3, 4, 5, 6)^T$. Thus, link 1 has the highest priority 1, and link 6 has the lowest priority 6. For link 1, its neighborhood is $\mathcal{N}_1 = \{2, 6\}$. Similarly, we have $\mathcal{N}_2 = \{1, 3\}$. The graph incidence matrix P associated with $\boldsymbol{p}$ is as follows: $P_{ii} = 1, 1 \le i \le 6$, $P_{i(i-1)} = 1, 2 \le i \le 6$, $P_{61} = 1$, and 0 at all the other entries. Let $\boldsymbol{a} = (0.3, 0.4, 0.3, 0.4, 0.3, 0.4)^T$ be an arrival

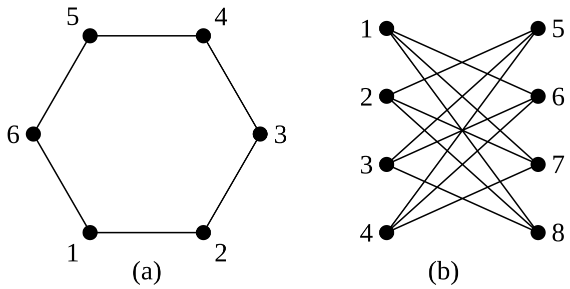

Fig. 1. (a) is a conflict graph consisting of 6 links, and (b) shows an incomplete bipartite graph.

rate vector. Clearly $\boldsymbol{a} \notin \mathcal{A}_{\text{maximal}}$, since $a_1 + \sum_{j\in\mathcal{N}_1} a_j = .3 + .4 + .4 = 1.1 > 1$. But we have $\boldsymbol{a} \in \mathcal{A}_{\boldsymbol{p}}$, since $\|P\boldsymbol{a}\|_\infty = \|(.3, .7, .7, .7, .7, 1)^T\|_\infty = 1$.

It is in general difficult to guarantee stability in a network, since the result has to hold over all arrival processes with the same average rate. However, characterizing the stability region is important in some applications, such as in cross-layer design, where the resources need to be allocated subject to the constraint of network stability. In next section we formulate a lower bound stability region of LQF scheduling.

III. Stability Region of LQF Scheduling

Section II-C argued the importance of stability region. For LQF scheduling, although the approximation ratio of LQF scheduling is well-known, and is equal to the "local-pooling ratio" of the network [5], the stability region of LQF scheduling is hard to describe. That is, given an arrival rate vector $\boldsymbol{a}$, it is difficult to predict that the network is stable under LQF scheduling without solving a problem of exponential complexity in n. Our previous work [6] on SP scheduling allows us to propose the following lower bound stability region of LQF scheduling: $\mathcal{A}_{\text{LQF}} = \{\boldsymbol{a} \in \mathbb{R}^n_+ : \min_{P\in\mathcal{P}} \|P\boldsymbol{a}\|_\infty < 1\}$, where $\mathcal{P}$ is the set of P matrices associated with $n!$ priority vectors. We remark that although $\boldsymbol{P}$ has $n!$ matrices, we show later in the section that optimization involved can be solved with linear complexity. The stability result is shown in the following theorem.

Theorem 1: The network is rate stable under LQF scheduling for any $\boldsymbol{a} \in \mathcal{A}_{\text{LQF}}$.

We first briefly describe the intuition behind the proof. Note that in LQF scheduling, links with longer queues are always considered before the links with shorter queues. So, queues that grow are approximately equal. Thus, if there is a set of links $\mathcal{V}_0$ which are the longest, LQF will guarantee that, in each time slot, the schedule is at least maximal when restricted to the links in $\mathcal{V}_0$. This, together with the fact that $\|P\boldsymbol{a}\|_\infty < 1$ for some $P \in \mathcal{P}$, guarantees that *some* queue in $\mathcal{V}_0$ is decreasing, which implies that the max queue length is also decreasing.

Proof: In the following proof we will use the technique of fluid limits [8] to prove rate stability. For a brief introduction about the derivation of fluid limits, please see Appendix.

Let a fluid limit $(\bar{\boldsymbol{Q}}(t), \bar{\boldsymbol{A}}(t), \bar{\boldsymbol{D}}(t))$ be given. Thus, there exists a sequence $\{r_n\}_{n=1}^\infty$, such that as $r_n \to \infty$, we have

$$(\frac{\boldsymbol{Q}(r_n t)}{r_n t}, \frac{\boldsymbol{A}(r_n t)}{r_n t}, \frac{\boldsymbol{D}(r_n t)}{r_n t}) \to (\bar{\boldsymbol{Q}}(t), \bar{\boldsymbol{A}}(t), \bar{\boldsymbol{D}}(t))$$

where the convergence is interpreted as uniformly on compact sets (u.o.c). Consider the Lyapunov function $L(\bar{\boldsymbol{Q}}(t)) = \max_{i\in\mathcal{V}} \bar{Q}_i(t)$, and let $t > 0$ be given. Now it is sufficient to argue that if $\boldsymbol{a} \in \mathcal{A}_{\text{LQF}}$, we have $\dot{L}(\bar{\boldsymbol{Q}}(t)) \le 0$, and therefore stability follows from Lemma 1.

At time t, denote $\mathcal{V}_0$ as the set of links with the longest queues in the fluid limit, i.e., with $\bar{Q}_i(t) = \max_{j\in\mathcal{V}} \bar{Q}_j(t)$. Thus, since the function $\bar{\boldsymbol{Q}}(t)$ is absolutely continuous, there exists $\epsilon > 0$ and $\delta > 0$ such that $\forall \tau \in (t-\delta, t+\delta)$, we have $\bar{Q}_i(\tau) - \bar{Q}_j(\tau) \ge \epsilon$ for any $i \in \mathcal{V}_0, j \in \mathcal{V}_0^c$. Thus, in the original network, for sufficiently large n we have

$$Q_i(r_n\tau) - Q_j(r_n\tau) \ge r_n\epsilon \ge 1, \forall (r_n\tau) \in (r_n(t-\delta), r_n(t+\delta))$$

for any $i \in \mathcal{V}_0, j \in \mathcal{V}_0^c$. Therefore, in the original network during the time interval $(r_n(t-\delta), r_n(t+\delta))$, none of the links in the set $\mathcal{V}_0$ has empty queue. Further, according to LQF, the links in $\mathcal{V}_0$ will always be considered by LQF before any link in $\mathcal{V}_0^c$. Thus, for any link $i \in \mathcal{V}_0$ and any $\tau \in (r_n(t-\delta), r_n(t+\delta))$, if none of link i's neighbor in $\mathcal{V}_0$ are scheduled, link i will be scheduled for transmission, i.e.,

$$\Delta D_i(\tau) + \sum_{j\in\mathcal{N}_i} \Delta D_j(\tau) \mathbf{1}_{\{j\in\mathcal{V}_0\}} \ge 1 \tag{4}$$

where $\Delta D(\tau) = D(\tau) - D(\tau - 1)$. After summing over the time interval $(r_n(t-\delta), r_n(t+\delta))$ we conclude that

$$\begin{aligned} & D_i(r_n(t+\delta)) + \sum_{j\in\mathcal{N}_i} D_j(r_n(t+\delta))\mathbf{1}_{\{j\in\mathcal{V}_0\}} \\ \ge\ & D_i(r_n(t-\delta)) + \sum_{j\in\mathcal{N}_i} D_j(r_n(t-\delta)))\mathbf{1}_{\{j\in\mathcal{V}_0\}} + 2r_n\delta \end{aligned}$$

from which we conclude that $\forall i$,

$$\dot{\bar{D}}_i(t) + \sum_{j\in\mathcal{N}_i} \dot{\bar{D}}_j(t)\mathbf{1}_{\{j\in\mathcal{V}_0\}} \ge 1 \tag{5}$$

Since $\boldsymbol{a} \in \mathcal{A}_{\text{LQF}}$, there exists $P \in \mathcal{P}$ such that $\|P\boldsymbol{a}\|_\infty < 1$. Denote $i^\star$ as the link in $\mathcal{V}_0$ with the lowest priority, we have

$$\begin{aligned} a_{i^\star} + \sum_{j\in\mathcal{N}_{i^\star}} a_j \mathbf{1}_{\{j\in\mathcal{V}_0\}} &\overset{(a)}{=} a_{i^\star} + \sum_{j\in\mathcal{N}_{i^\star}} a_j \mathbf{1}_{\{j\in\mathcal{V}_0\}}\mathbf{1}_{\{p_{i^\star} > p_j\}} \\ &\le a_{i^\star} + \sum_{j\in\mathcal{N}_{i^\star}} a_j \mathbf{1}_{\{p_{i^\star} > p_j\}} \le 1 \end{aligned} \tag{6}$$

where (a) is because link $i^\star$ has the lowest priority in $\mathcal{V}_0$. Thus, we have

$$\begin{aligned} & \dot{\bar{Q}}_{i^\star}(t) + \sum_{j\in\mathcal{N}_{i^\star}} \dot{\bar{Q}}_j(t)\mathbf{1}_{\{j\in\mathcal{V}_0\}} \\ \overset{(a)}{=}\ & a_{i^\star} + \sum_{j\in\mathcal{N}_{i^\star}} a_j - (\dot{\bar{D}}_{i^\star}(t) + \sum_{j\in\mathcal{N}_i} \dot{\bar{D}}_j(t)\mathbf{1}_{\{j\in\mathcal{V}_0\}}) \\ \overset{(b)}{\le}\ & a_{i^\star} + \sum_{j\in\mathcal{N}_{i^\star}} a_j - 1 \le 0 \end{aligned}$$

where (a) is because of SLLN and that link $i^\star$ has the lowest priority in $\mathcal{V}_0$, and (b) is because of (5). Thus, (6) shows that $\dot{\bar{Q}}_{i^\star}(t) \leq 0$. Note that for any regular $t > 0$, we have $\dot{\bar{Q}}_j(t) = \dot{\bar{Q}}_{i^\star}(t)$ for all $j \in \mathcal{V}_0$. Therefore, we conclude that $\dot{L}(\bar{\boldsymbol{Q}}(t)) = \dot{\bar{Q}}_{i^\star}(t) \leq 0$ and the theorem follows by applying Lemma 1. ∎

We next show the tightness of the stability region $\mathcal{A}_{\mathrm{LQF}}$ in some scenarios. We will use the example proposed in [5]. Consider the 6-ring network in Fig. 1, and let the arrival rate be $\boldsymbol{a} = (1/3 + \epsilon)\mathbf{1}$, where $\mathbf{1}$ is the all-ones vector. Define a periodic arrival process as follows: At time slot $3k+1$, one packet arrives at link 1 and 4, and at time slot $3k+2$, one packet arrives at link 2 and 5, and at time slot $3k+3$, one packet arrives at link 3 and 6. Additionally, in each time slot, with probability ϵ, one packet arrives at each and every link in the network. Note that LQF will always schedule $(1,4)$, $(2,5)$ and $(3,6)$ and all the queues grow unbounded with rate ϵ. Therefore, for this network there is an arrival rate vector which is arbitrarily close to $\mathcal{A}_{\mathrm{LQF}}$, which can not be stabilized by LQF scheduling.

Note that even if the close-form stability region $\mathcal{A}_{\mathrm{LQF}}$ is given, testing whether $\boldsymbol{a} \in \mathcal{A}_{\mathrm{LQF}}$ is still a nontrivial problem since one needs to consider $n!$ priorities. However, we now show that we can test whether $\boldsymbol{a} \in \mathcal{A}_{\mathrm{LQF}}$ efficiently. In fact, the following algorithm Test-Feasibility, can achieve this with only linear complexity.

Algorithm 1 Test-Feasibility $(\mathcal{G}, \boldsymbol{a})$

```
for k = n to 1 do
  s_k = arg min_{i∈V} {a_i + Σ_{j∈N_i} a_j};
  if (a_{s_k} + Σ_{j∈N_{s_k}} a_j > 1) then
    return FALSE;
  else
    Remove link s_k and its incident edges from G;
  end if
end for
return TRUE;
```

We finally conclude this section with the following theorem.

Theorem 2: Algorithm TF will return TRUE if and only if $\boldsymbol{a} \in \mathcal{A}_{\mathrm{LQF}}$.

Due to space limit, we only describe the intuition behind the proof. The formal proof is similar to the proof of Theorem 3 in [6]. Essentially, if any link i satisfies $a_i + \sum_{j\in\mathcal{N}_i} a_j \leq 1$, we say that link i is worst-case stable, i.e., for any priority vector $\boldsymbol{p}$, we have $(P\boldsymbol{a})_i \leq 1$. Thus, if we assign i the lowest priority, $(P\boldsymbol{a})_i = a_i + \sum_{j\in\mathcal{N}_i} a_j \leq 1$, and for any $j \neq i$, $(P\boldsymbol{a})_j$ can only get smaller since i now has the lowest priority. Thus, we are not losing optimality by reassigning the lowest priority to a worst-case stable link i. The proof follows by repeating the above arguments.

IV. Static Priority Scheduling

The stability region of LQF scheduling, $\mathcal{A}_{\mathrm{LQF}}$, can be improved by using SP scheduling, as this section shows. In this section we consider SP scheduling with multiple priorities, which are parameterized by $\{\boldsymbol{p}^{(k)}, \boldsymbol{a}^{(k)}, \theta^{(k)}\}_{k=1}^K$, where $\sum_{k=1}^K \boldsymbol{a}^{(k)} = \boldsymbol{a}$ and $\sum_{k=1}^K \theta^{(k)} = 1$. The scheduling algorithm is described as follows: We divide time slots into blocks where each block with length T consists of k sub-blocks, such that the k-th block has a length of $\theta^{(k)}T$. Further, each link $i \in \mathcal{V}$ has K sub-queues where each sub-queue k has arrival rate $a_i^{(k)}$, so that $\sum_{k=1}^K a_i^{(k)} = a_i$. Note that this can be achieved by filtering the arrival processes probabilistically into K sub-queues. During the scheduling in each k-th time block, only the sub-queues indexed by k are allowed for transmission, which follows the order as specified by priority vector $\boldsymbol{p}^{(k)}$. We have the following theorem about the stability region.

Theorem 3: The network is rate stable under the SP scheduling as described above if

$$\|P^{(k)}\boldsymbol{a}^{(k)}\|_\infty < \theta^{(k)}, \forall 1 \leq k \leq K \tag{7}$$

where $P^{(k)}$ is the incidence matrix associated with $\boldsymbol{p}^{(k)}$.

Proof: Let $1 \leq k \leq K$ be given, and consider any fluid limit $(\bar{\boldsymbol{Q}}^{(k)}(t), \bar{\boldsymbol{A}}^{(k)}(t), \bar{\boldsymbol{D}}^{(k)}(t))$ with a converging sequence $\{(\bar{\boldsymbol{Q}}^{(k)r_n}(t), \bar{\boldsymbol{A}}^{(k)r_n}(t), \bar{\boldsymbol{D}}^{(k)r_n}(t))\}_{n=1}^\infty$. We will argue that every sub-queue is stable, i.e., $\bar{\boldsymbol{Q}}^{(k)}(t) = \mathbf{0}$ for all $t \geq 0$ if $\bar{\boldsymbol{Q}}^{(k)}(t) = \mathbf{0}$, and therefore stability follows from Lemma 1. For the k-th sub-queues, define Lyapunov function $L(\bar{\boldsymbol{Q}}^{(k)}(t)) = \frac{1}{2}\|\bar{\boldsymbol{Q}}^{(k)}(t)\|_2^2$, and consider the link i_1 with the highest priority according to $\boldsymbol{p}^{(k)}$. Suppose that $\bar{Q}_{i_1}^{(k)}(t) \geq \epsilon > 0$ at time $t > 0$. Then there exists $\delta > 0$ such that $\bar{Q}_{i_1}^{(k)}(\tau) \geq \epsilon/2$ for $\tau \in (t-\delta, t+\delta)$. Thus, for sufficiently large n, we have $Q_{i_1}^{(k)}(r_n\tau)/r_n\tau \geq \epsilon/4$ for $(r_n\tau) \in (r_n(t-\delta), r_n(t+\delta))$, i.e., if we choose n large enough, $Q_{i_1}^{(k)}(\tau)$ is never empty during the time interval $(r_n(t-\delta), r_n(t+\delta))$. Thus, according to SP scheduling, there is $\theta^{(k)}T$ packet departures from $Q_{i_1}^{(k)}(\tau)$ in every time block with length T, and we conclude that $\frac{d}{dt}\bar{D}_i^{(k)}(t) = \theta^{(k)}$, and therefore

$$\frac{d}{dt}\frac{1}{2}\bar{Q}_{i_1}^{(k)2}(t) = \bar{Q}_{i_1}^{(k)}(t)\dot{\bar{Q}}_{i_1}^{(k)}(t) = \bar{Q}_{i_1}^{(k)}(t)(a_{i_1}^{(k)} - \theta^{(k)}) \overset{(a)}{\leq} 0$$

where (a) is because $a_{i_1}^{(k)} = (P^{(k)}\boldsymbol{a}^{(k)})_{i_1} \leq \|P^{(k)}\boldsymbol{a}^{(k)}\|_\infty \leq \theta^{(k)}$ since i_1 has the highest priority according to $\boldsymbol{p}^{(k)}$. Thus, we conclude that $\bar{Q}_{i_1}^{(k)}(t) = 0$ for all $t \geq 0$.

Now suppose this is true for the links $i_1, i_2, \ldots, i_l$, i.e., the l links with the highest priorities. Now consider link i_{l+1}, and suppose that $\bar{Q}_{i_{l+1}}^{(k)}(t) > 0$ at some time $t > 0$. Using similar arguments, we have

$$\begin{aligned}\frac{d}{dt}\frac{1}{2}\bar{Q}_{i_{l+1}}^{(k)2}(t) &= \bar{Q}_{i_{l+1}}^{(k)}(t)\dot{\bar{Q}}_{i_{l+1}}^{(k)}(t) \\ &\overset{(a)}{=} \bar{Q}_{i_{l+1}}^{(k)}(t)(\dot{\bar{Q}}_{i_{l+1}}^{(k)}(t) + \sum_{j\in\mathcal{N}_i} \dot{\bar{Q}}_{i_j}^{(k)}(t)\mathbf{1}_{\{p_i^{(k)} > p_j^{(k)}\}}) \\ &= \bar{Q}_{i_{l+1}}^{(k)}(t)(a_{i_{l+1}}^{(k)} + \sum_{j\in\mathcal{N}_i} a_j^{(k)}\mathbf{1}_{\{p_i^{(k)} > p_j^{(k)}\}} - \theta^{(k)}) \\ &\leq 0\end{aligned}$$

where (a) is because $\bar{Q}_{i_j}^{(k)}(t)=0$ for links $1\leq j\leq l$. Thus, induction shows that $\bar{Q}_{i_{l+1}}^{(k)}(t)=0$ for all $t>0$, and therefore, the network is rate stable. ∎

We have the following corollary, which says that it is sufficient to consider less than $n+1$ priorities to achieve the optimal stability region $\mathcal{A}_{\max}$ [1], which is the convex hull of the set of independent sets.

Corollary 1: Let $\boldsymbol{a}\in\mathcal{A}_{\max}$ be given. For $K=n+1$, there exists a SP scheduling with parameters $(\boldsymbol{p}^{(k)},\boldsymbol{a}^{(k)},\theta^{(k)})_{k=1}^{K}$ such that $\boldsymbol{a}$ is stable.

Proof: Note that this is similar to the statement in [7]. We briefly provide the proof here for completeness. Since $\boldsymbol{a}\in\mathcal{A}_{\max}$, from Carathéodory Theorem, we can express $\boldsymbol{a}$ as a convex combination of at most $n+1$ independent sets, i.e., $\boldsymbol{a}=\sum_{k=1}^{n+1}\theta^{(k)}\boldsymbol{m}^{(k)}$, where $\boldsymbol{m}^{(k)}$ is an $n\times 1$ indicator vector representing an independent set, i.e., $m_i^{(k)}=1$ if link i is included in the set, otherwise $m_i^{(k)}=0$. Now define $\boldsymbol{a}^{(k)}=\theta^{(k)}\boldsymbol{m}^{(k)}$, and choose priority $\boldsymbol{p}^{(k)}$ such that the links in the independent set $\boldsymbol{m}^{(k)}$ have locally highest priority. Thus, $P^{(k)}$ is a diagonal matrix and $P^{(k)}\boldsymbol{a}^{(k)}=\boldsymbol{a}^{(k)}$. With this choice of $(\boldsymbol{p}^{(k)},\boldsymbol{a}^{(k)},\theta^{(k)})_{k=1}^{K}$, it is easily seen that (7) is satisfied and the network is stable. ∎

Although the lower bound stability region for SP scheduling is known, the assignment of parameters $(\boldsymbol{p}^{(k)},\boldsymbol{a}^{(k)},\theta^{(k)})_{k=1}^{K}$ is not trivial. For the special case with $K=1$ and $K=n+1$, we have shown the optimal priority assignment algorithms [7]. For the cases $2\leq k\leq K$, it is, in general, hard to assign the parameters optimally. In the following we consider a special case, where we assume two priorities $\boldsymbol{p}^{(1)},\boldsymbol{p}^{(2)}$ and $\theta^{(1)}=\theta^{(2)}=1/2$, for the ease of implementation. In this case, we can obtain the priorities by solving the following problem:

$$\begin{aligned}\text{OPT:}\quad &\min_{P^{(1)},P^{(2)},\boldsymbol{x}} && \max(\|P^{(1)}\boldsymbol{x}\|_\infty,\|P^{(2)}(\boldsymbol{a}-\boldsymbol{x})\|_\infty)\\ &\text{subject to} && \boldsymbol{0}\preceq\boldsymbol{x}\preceq\boldsymbol{a}\\ & && P^{(1)},P^{(2)}\in\mathcal{P}\end{aligned}$$

Note that even in this special case, an optimal solution is hard to get since the problem is non-convex. We next propose a heuristic algorithm (Alg. 2) to solve the above problem, which is related to the well-known EM algorithm in the literature.

After each iteration, the objective function value in OPT gets smaller, and therefore converges to a local optimal assignment. If the limit of t's is less than 1, $\boldsymbol{a}$ is stable using the 2 priority vectors. We will test the performance of the EM algorithm in the next section.

V. Simulation Results

In this section we demonstrate the performance of LQF and SP scheduling through simulation. Since the performance of LQF and SP scheduling is dependent on the arrival processes as well as the network topology, an exhaustive search of arrival processes and the networks is clearly not possible. Thus, similar to the recent research [10], where specific graph structures are explored to demonstrate the performance limit of scheduling, we will consider special network topologies and arrival processes to demonstrate the performance limit of LQF and SP scheduling.

Algorithm 2 EM$(\mathcal{P},\boldsymbol{a})$

while (1) **do**

1) E-Step: Choose priorities $(P^{(1)\star},P^{(2)\star})$ by solving

$$\begin{aligned}&\min_{P^{(1)},P^{(2)}\in\mathcal{P}}\max(\|P^{(1)}\boldsymbol{x}^\star\|_\infty,\|P^{(2)}(\boldsymbol{a}-\boldsymbol{x}^\star)\|_\infty)\\ =\;&\max(\min_{P^{(1)}\in\mathcal{P}}\|P^{(1)}\boldsymbol{x}^\star\|_\infty,\min_{P^{(2)}\in\mathcal{P}}\|P^{(2)}(\boldsymbol{a}-\boldsymbol{x}^\star)\|_\infty)\end{aligned}$$

2) M-Step: Choose arrival rates $(\boldsymbol{x}^\star,\boldsymbol{a}-\boldsymbol{x}^\star)$ by solving

$$\begin{aligned}&\min_{\boldsymbol{x},t} && t\\ &\text{subject to} && P^{(1)\star}\boldsymbol{x}\preceq(t/2)\mathbf{1}\\ & && P^{(2)\star}(\boldsymbol{a}-\boldsymbol{x})\preceq(t/2)\mathbf{1}\\ & && \boldsymbol{0}\preceq\boldsymbol{x}\preceq\boldsymbol{a}\end{aligned}$$

return if the sequence of t's has converged

end while

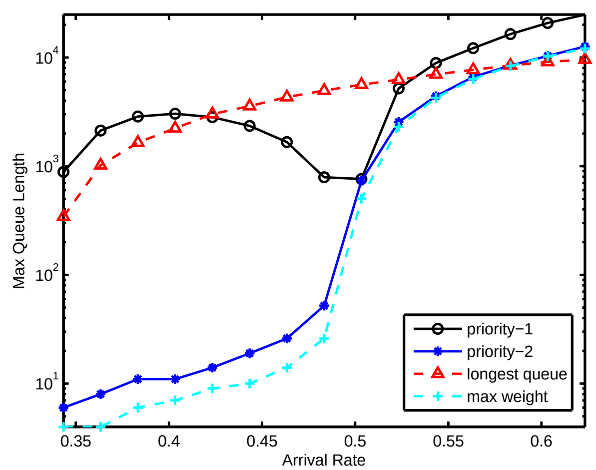

Fig. 2. The simulation result of a 6-ring network with different schedulers.

A. 6-Ring

We first consider the 6-ring network in Fig. 1(a) and the arrival process as described in Section III ($\boldsymbol{a}$ is uniform). Fig. 2 shows the network stability result with respect to the uniform arrival rate over a time period of 10^5 time slots, where the result is averaged over 10 independent simulations. Note that boundary of the optimal stability region is at a uniform arrival rate of 0.5, above which the clique constraint (i.e., a single edge) is violated. One can observe that neither the LQF scheduling nor the SP scheduling with a single priority vector is stable, as the max queue length is large (the decreasing behavior of SP scheduling with single priority near 0.5 is due to the specific arrival process) However, both the Max-Weight scheduling [1] and SP scheduling with two (optimal) priority vectors can stabilize the network.

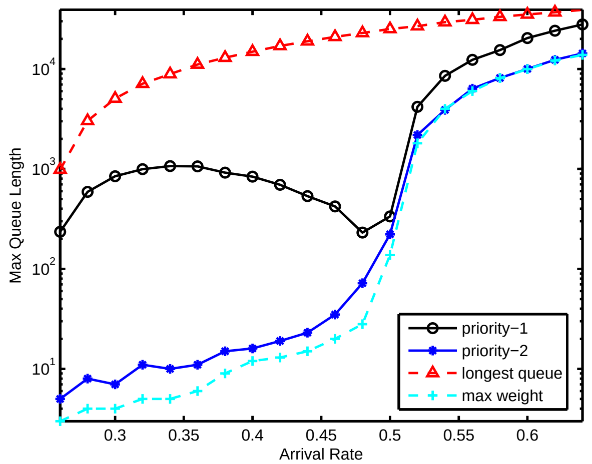

Fig. 3. The simulation result of a bipartite network with different schedulers.

B. Bipartite Graph

We next consider an incomplete bipartite graph with 8 links as shown in Fig. 1(b). We consider a periodic arrival process which is similar to the one for the 6-ring network. The stability result is shown in Fig. 3. Similarly, one can observe that the LQF scheduling and SP scheduling with single priority vector is not stable, whereas both the SP scheduling with two priorities and the max-weight scheduling can stabilize the maximum uniform arrival rate, which is 0.5 for this network.

VI. Conclusion

In this paper we considered two greedy scheduling algorithms in wireless networks: LQF scheduling and SP scheduling. For LQF scheduling, we formulated a close-form stability region, which was shown to be tight in some scenarios. For SP scheduling with multiple priorities, we also proposed a lower bound stability region, as well as a heuristic priority assignment algorithm, which is related to the well-known EM algorithm. The performance gain of the proposed heuristic algorithm was finally confirmed by simulations.

Appendix

In the appendix we introduce fluid limits [8], which are used to prove rate stability.

A. Existence of Fluid Limits

Given the network dynamics which are described by the functions $(\boldsymbol{Q}(t), \boldsymbol{A}(t), \boldsymbol{D}(t))_{t=0}^{\infty}$, we first extend the support from $\mathbb{N}$ to $\mathbb{R}_+$ using linear interpolation. Now, for a fixed sample path ω, define the fluid scaling of a function $f(t)$ as $f^r(t) = f(rt)/r$, where f can be $Q_i(t), A_i(t)$ or $D_i(t)$. It can be verified that these functions are uniformly Lipschitz-continuous, i.e., for any $t > 0$ and $\delta > 0$, we have

$$\begin{aligned} A_i^r(t+\delta) - A_i^r(t) &\leq A_{\max}\delta && (8)\\ D_i^r(t+\delta) - D_i^r(t) &\leq \delta && (9)\\ Q_i^r(t+\delta) - Q_i^r(t) &\leq A_{\max}\delta && (10)\end{aligned}$$

Thus, these functions are equi-continuous. According to Arzéla-Ascoli Theorem, any sequence of functions $\{f^{r_n}(t)\}_{n=1}^{\infty}$ contains a subsequence $\{f^{r_{n_k}}(t)\}_{k=1}^{\infty}$, such that such that

$$\lim_{k\to\infty} \sup_{\tau\in[0,t]} |f^{r_{n_k}}(\tau) - \bar{f}(\tau)| = 0 \qquad (11)$$

where $\bar{f}(t)$ is a uniformly continuous (and therefore differentiable almost everywhere) function. In our example, $f(t)$ can be $Q_i(t), A_i(t)$ or $D_i(t)$. Define any such limit $(\bar{\boldsymbol{Q}}(t), \bar{\boldsymbol{A}}(t), \bar{\boldsymbol{D}}(t))$ as a fluid limit.

B. Properties of Fluid Limits

We have the following properties holds for any fluid limit

$$\begin{aligned} \bar{A}_i(t) &= a_i t && (12)\\ \frac{d}{dt}\bar{Q}(t) &= 0 \quad \text{if } \bar{Q}(t) = 0 && (13)\end{aligned}$$

where (12) is because of (functional) SLLN, and (13) is because any regular point t with $\bar{Q}(t) = 0$ achieves local minimum (since $\bar{Q}(t) \geq 0$), and therefore has zero derivative. We further have the following lemma which states a sufficient condition about rate stability [8]:

Lemma 1: The network is rate stable if any fluid limit with $\bar{\boldsymbol{Q}}(0) = \mathbf{0}$ has $\bar{\boldsymbol{Q}}(t) = \mathbf{0}, \forall t \geq 0$.

References

[1] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. on Automatic Control*, Vol. 37, No. 12, pp. 1936-1949, December 1992

[2] L. Tassiulas, "Linear Complexity Algorithms for Maximum Througput in Radio Networks and Input Queued Switches," *Proc. of IEEE INFOCOM*, 533-539, 1998

[3] L. Jiang and J. Walrand, "A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks", Allerton Conference on Communication, Control, and Computing, 2008

[4] P. Chaporkar, K. Kar, X. Luo, and S. Sarkar, "Throughput and Fairness Guarantees Through Maximal Scheduling in Wireless Networks", *IEEE Trans. on Info. Theory*, Vol. 54, No. 2, February 2008, pp. 572-594.

[5] C. Joo, X. Lin, and N. B. Shroff, "Understanding the Capacity Region of the Greedy Maximal Scheduling Algorithm in Multi-hop Wireless Networks," *IEEE INFOCOM*, Phoenix, Arizona, April 2008

[6] Q. Li and R. Negi, "Prioritized Maximal Scheduling in Wireless Networks", *Proc. of IEEE Globecom*, 2008

[7] Q. Li and R. Negi, "Scheduling in Wireless Multi-hop Networks with Priorities", *Proc. of IEEE Infocom*, 2009

[8] J.G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," *Proc. of the IEEE INFOCOM*, 2:556-564, March 2000.

[9] A. Dimakis and J. Walrand, "Sufficient Conditions for Stability of Longest Queue First Scheduling: Second Order Properties Using Fluid Limits," *Journal of Applied Probability*, May 2005.

[10] B. Birand, M. Chudnovsky, B. Ries, P. Seymour, G. Zussman and Y. Zwols, "Analyzing the Performance of Greedy Maximal Scheduling via Local Pooling and Graph Theory," *Proc. IEEE INFOCOM* 2010, Mar. 2010